WE CLAIM:

1. A method of segmentation of variable-size packets, each packet being associated with a defined data stream, wherein packets of a defined data stream are concatenated and divided into equal-size segments, the packet concatenation and segmentation being constrained by adaptive delay thresholds.
2. The method as claimed in claim 1 wherein the adaptive-delay thresholds are determined by allocated transfer rates.
3. The method as claimed in claim 2 wherein the adaptive delay threshold is governed by a transfer rate controller.
4. The method as claimed in claim 2 wherein the magnitude of the adaptive delay threshold is data-stream dependent.
5. The method as claimed in claim 1 wherein packet concatenation is constrained by data storage limitation.
6. The method as claimed in claim 1 wherein a data stream comprises packets from a multiplicity of traffic sources.
7. In a network comprising source nodes and sink nodes, the method as claimed in claim 1 wherein said concatenation is applied at the ingress port of a source node.
8. In a network comprising source nodes and sink nodes, the method as claimed in claim 1 wherein said concatenation is applied at the output port of a source node.
9. The method as claimed in claim 1 wherein the number of packet membership per segment is limited by a predefined upper bound.
10. In a switching node receiving packets from a plurality of incoming channels, a method of two-phase packet segmentation wherein the first phase segments individual packets and the second phase segments an aggregate of packets.
11. The method as claimed in claim 10 wherein the aggregate of waiting packets contains only one packet.

12. The method as claimed in claim 10 wherein the size of an output segment of the first phase is smaller than the size of an output segment of the second phase.

13. The method as claimed in claim 10 wherein the size of a second-phase segment is an integer multiple of the size of a first-phase segment.

14. The method as claimed in claim 10 wherein the waiting packets are received from a single incoming channel.

15. The method as claimed in claim 10 wherein the waiting packets are received from a plurality of incoming channels.

16. The method as claimed in claim 15 wherein each packet is associated with an identified data stream and the waiting packets are sorted according to the associated-stream identifiers.

17. The method as claimed in claim 10 wherein the output of the first phase is switched to the second phase.

18. The method as claimed in claim 10 wherein said first-phase segmentation is implemented by a plurality of first-phase segmentation circuits and said second-phase segmentation is implemented by a plurality of second-phase segmentation circuits.

19. The method as claimed in claim 18 wherein a first-phase segmentation circuit is allocated to an incoming channel.

20. The method as claimed in claim 18 wherein the output of said first-phase segmentation circuit is switched to said second-phase segmentation circuits under control of a routing algorithm.

21. The method as claimed in claim 20 wherein the routing algorithm selects routes from an incoming channel to a sink node using a method of maximum aggregation probability.

22. A data structure to enable merging equal-size null-padded data segments into new equal-size data segments having reduced null-padding, the data segments belonging to $K > 1$ data streams, the data structure comprising:

(i) An array "A" of fractional segments having K entries, each entry dedicated to a data stream;

(ii) An array "B" of complete segments having K1 > K entries; and

(iii) a control matrix "C" having K records, each record dedicated to a data stream and having four fields to indicate, for a corresponding stream, the stream's delay threshold, the stream's current delay, the fill of a respective fractional-segment in array "A", and the address of a complete segment in array "B".

23. The data structure as claimed in claim 22 wherein the size of each of said new equal-size data segments is an integer-multiple of the size of each of said equal-size null-padded data segments.

24. The data structure as claimed in claim 22 wherein array "A" is directly indexed by a stream identifier.

25. The data structure as claimed in claim 22 wherein array "A" is replaced by a pointer array and a segment-storage array.

26. In conjunction with the data structure as claimed in claim 22, a method for data segmentation including steps of

(a) Receiving a plain segment of a stream k, $0 \leq k < K$;

(b) Merging said plain segment with data content of array "A" at storage position k;

(c) Appending a complete segment resulting from said merging to a queue associated with data stream k in array "B"; and

(d) Overwriting a fractional segment resulting from said merging in memory "A" at position k.

27. In conjunction with the data structure as claimed in claim 22, a method of controlling data segments transfer, the method including steps of:

    (a) Receiving from a transfer rate controller a prompt to transfer a data segment of a stream k, $0 \leq k < K$;

    (b) Transferring a complete segment belonging to data stream k in array "B" if said complete segment exists; else

    (c) Transferring a fractional segment belonging to data stream k in array "A" if said fractional segment exists and has already been prompted by said rate controller a predefined number of times; else

    (d) Return a no-action indication to rate controller.

28. The method as claimed in claim 26 wherein a plurality of plain segments are processed concurrently.

29. The method as claimed in claim 26 wherein pointers to transferred segments are placed in a transfer queue.

30. A compact-segmentation apparatus operable to segment variable-size packets into compact fixed-size segments, the apparatus comprising:

    (a) a packet segmentation circuit;
    (b) an enqueueing controller;
    (c) a principal data memory;
    (d) an auxiliary data memory;
    (e) a control memory;
    (f) a dequeueing controller; and
    (g) a transfer-rate controller;

wherein

said packet segmentation circuit receives variable-size packets and segments each of said variable-size packets into plain segments,

said packet segmentation circuit identifies a data stream associated with each one of said variable-size packets,

said enqueueing controller communicates with said control memory, said principal data memory, and said auxiliary data memory, and concatenates said plain segments of a data stream with previously-received segments of same data stream,

said transfer-rate controller selects a selected data stream, and

said dequeueing controller selects data segments, belonging to said selected data stream, from waiting data segments in said principal data memory and said auxiliary data memory for transfer downstream.

31. The compact packet segmentation circuit as claimed in claim 30 wherein the rate controller is driven by a service-quality controller.

32. The compact packet segmentation circuit as claimed in claim 31 wherein the service-quality controller is source-node driven.

33. The compact packet segmentation circuit as claimed in claim 31 wherein the service-quality controller is driven by a source node controller.

34. The compact packet segmentation circuit as claimed in claim 30 wherein the principal data memory is structured as a multi-head interleaved linked list.

35. The apparatus as claimed in claim 31 wherein a segment is ready for transfer by the dequeueing controller if it is a complete segment waiting in the principal data memory or a fractional segment waiting in the auxiliary data memory and has already been prompted by the rate controller a predefined number of times.

36. A circuit for fast concatenation of a first packet of a known length and a second packet of a known length into a complete segment of a predefined length, the circuit comprising a shift connector, a memory array, and a register array,

wherein the length of said first and second packets is smaller than said predefined length,

and wherein said first packet is copied onto said register array, said shift connector performs a concatenation process that concatenates said second packet with said first

packet onto said register array placing any remainder resulting from said concatenation process in said memory array.

37. The circuit as claimed in claim 36 wherein the shift connector is constructed from arrays of shift connectors of a smaller size.

38. The circuit as claimed in claim 37 wherein each array comprises identical shift connectors and the sizes of shift connectors in different arrays vary.

39. **A** heterogeneous data segment format comprising a front header followed by a corresponding payload data and a plurality of inner headers, each inner header followed by a corresponding payload data.

40. The heterogeneous data segment format as claimed in claim 39 wherein the front header further includes a field indicating a respective network address and a field indicating length of said corresponding payload data .

41. The heterogeneous data segment format as claimed in claim 39 wherein each inner header includes only one field indicating a length of corresponding payload data.

42. The heterogeneous data segment format as claimed in claim 39 wherein the number of inner headers is limited by an upper bound.

Patent Pending